

"Express Mail" Mailing Label No. EV 314901728 US

Date of Deposit March 31, 2004

Case No. 9683/184

5

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
APPLICATION FOR UNITED STATES LETTERS PATENT

10

INVENTORS:

Yuichi ICHIKAWA  
Naoki NARUSE  
Tatsuro OI  
Nobuyuki WATANABE  
Yasunori HATTORI  
Masato TAKESHITA  
Masakazu NISHIDA  
Mao ASAI  
Masayuki TSUDA  
Atsuki TOMIOKA  
Kazuhiro YAMADA  
Dai KAMIYA  
Satoshi WASHIO  
Naoki YAMANE  
Keiichi MURAKAMI

TITLE:

COMMUNICATION DEVICE AND  
PROGRAM

ATTORNEY:

Tadashi Horie  
(Registration No. 40,437)  
BRINKS HOFER GILSON & LIONE  
POST OFFICE BOX 10395  
CHICAGO, ILLINOIS 60610  
(312) 321-4200

## COMMUNICATION DEVICE AND PROGRAM

5       **[0001]**     This application claims priority under 35 U.S.C. §119 to Japanese Patent Application No. 2003-097143 filed March 31, 2003, the entire content of which is hereby incorporated by reference.

## TECHNICAL FIELD

**[0002]**     The present invention relates to a method for using contents.

## BACKGROUND ART

10       **[0003]**     In recent years, various systems have been developed which make it possible for a user of a communication device to download via the Internet, a variety of contents such as game software, from a server. In such contents downloading systems, when a user instructs a communication device to download contents, the contents are downloaded to the device and stored in the memory of the device. After storing the contents, it is possible for a user to use the contents.

15       **[0004]**     However, there is a drawback in the above described systems, in that, a user does not have the option of trying out contents to know what they are like before downloading and storing the contents on a communication device. This becomes problematic in the case of contents being unacceptable to the user, who has to necessarily download and store the contents and carryout a delete operation if the contents are found to be unacceptable.. The above system of operations may be felt as burdensome for a user who wishes to only try out contents.

20       **[0005]**     Japanese Patent Application Laid-Open Publication No. 63-14258 discloses a computer system, which upon receiving an application program (contents) from a host computer, automatically starts the application without having to receive an instruction from a user to start the application program.

25       **[0006]**     According to the computer system disclosed in the publication, a user can try out contents readily; however, a user must still direct the deletion of the contents.

## SUMMARY OF THE INVENTION

**[0007]** The present invention has been made in view of the circumstances stated above, and provides a method for making it more convenient for a user to try out contents.

5 **[0008]** To solve the above problems, the present invention provides a communication device comprising: operation means; cache memory means; content storage means composed of nonvolatile memory; receiving means for receiving contents; first writing means, when the receiving means receives contents, for writing the contents in a free space or a space, where some data is already stored, of the cache memory means; content using means, after the first  
10 writing means writes contents in the cache memory means, for processing or executing the contents; and second writing means, when a command is issued using the operation means for storing contents processed or executed by the content using means, for writing the contents in the content storage means after  
15 reading the contents from the cache memory means.

**[0009]** In one preferred embodiment, the receiving means may receive trial information indicating that the contents are contents for trial use; and the first writing means, when the receiving means receives the trial information, may write the contents in the free space or the space, where some data is already stored, of  
20 the cache memory means.

**[0010]** In another preferred embodiment, the second writing means may comprise determining means for determining whether a size of data readable in a free space of the content storage means is equal to or more than a data size of the contents stored in the cache memory means; and when the determining means  
25 determines that the size of data readable in the free space of the content storage means is equal to or more than the size of data of the contents stored in the cache memory means, the second writing means may write in the content storage means the contents processed or executed by the content using means after reading the contents from the cache memory means.

[0011] In another preferred embodiment, the writing means may comprise inquiring means, when the determining means determines that the size of data readable in the size of free space of the content storage means is smaller than the size of data of the contents stored in the cache memory means, for prompting a user to delete one or more contents stored in the content storage means; and when, in response to the prompting of the inquiring means, a command is issued by using the operation means for deleting one or more contents stored in the content storage means, the second writing means may, reserve a free space in the content storage means by making it impossible to read the contents from the content storage means and making a space in the content storage means storing the contents available for storing new data, and cause the determining means to make a determination.

[0012] The present invention provides a program product for causing a computer having operation means, cache memory means, and content storage means composed of nonvolatile memory, to execute: a receiving process of receiving contents; a first writing process, when contents are received in the receiving process, of writing the contents in a free space or a space where some data is already stored, of the cache memory means; a content using process, after contents are written in the cache memory means in the first writing process, of processing or executing the contents; and a second writing process, when a command is issued using the operation means for storing contents processed or executed in the content using process, of writing the contents in the content storage means after reading the contents from the cache memory means. The program may be provided through a computer readable storage medium storing the program.

[0013] In the present invention, when contents are received, the contents are written in a free space or a space where some data is already stored, of cache memory means. When contents are written in the cache memory means, the contents are processed or executed. When a command is issued using operation means for storing contents processed or executed, the contents are read from the cache memory means, and written in content storage means.

[0014] The present invention makes it more convenient for a user to try out contents.

#### [0015] BRIEF DESCRIPTION OF THE DRAWINGS

5 [0016] Fig. 1 is a block diagram showing an example of the configuration of communication system 1 according to an embodiment of the present invention.

[0017] Fig. 2 is a diagram showing an example of a list page displayed in liquid crystal display unit 405 of mobile phone 40 according to the embodiment.

[0018] Fig. 3 is a diagram showing the contents of an explanatory file according to the embodiment.

10 [0019] Fig. 4 is a diagram showing an example of an explanatory page displayed in liquid crystal display unit 405 of mobile phone 40 according to the embodiment.

[0020] Fig. 5 is a diagram showing an example of the data configuration of an ADF according to the embodiment.

15 [0021] Fig. 6 is a block diagram showing an example of the hardware configuration of mobile phone 40 according to the embodiment.

[0022] Fig. 7 is a block diagram showing an example of the Java Runtime Environment according to the embodiment.

20 [0023] Fig. 8 is a flowchart illustrating a speedy execution process performed by CPU 400 of mobile phone 40 according to the embodiment.

[0024] Fig. 9 is a flowchart illustrating a storage process performed by CPU 400 of mobile phone 40 according to the embodiment.

[0025] Fig. 10 is a flowchart illustrating a space freeing process performed by CPU 400 of mobile phone 40 according to the embodiment.

#### 25 DESCRIPTION OF PREFERRED EMBODIMENTS

[0026] Following is a description of one preferred embodiment according to the present invention with reference to the attached drawings. In the drawings, like elements are denoted by like numerals. It is intended that the preferred

embodiment be regarded as illustrative rather than limiting, and can be changed within the scope of the present invention.

**[0027]** 1. Configuration

Configuration of Communication System 1:

Fig. 1 is a block diagram showing the configuration of communication system 1 according to the embodiment.

**[0028]** Mobile packet communication network 30 is a communication network for providing mobile packet communication services to mobile phone 40 served by mobile packet communication network 30. Mobile packet communication network 30 comprises gateway server 31 and base station 32. In a communication service area of mobile packet communication network 30, many base stations 32 are provided. Mobile phone 40, when located in a radio cell covered by base station 32, can perform radio communication with base station 32.

**[0029]** Gateway server 31 relays data between mobile packet communication network 30 and Internet 20.

**[0030]** Mobile phone 40 and content server 10 perform Hyper Text Transfer Protocol (HTTP) communication via Internet 20 and mobile packet communication network 30.

**[0031]** Configuration of Content Server 10:

Following is a description of the configuration of content server 10. The configuration of content server 10 is similar to that of common computers, therefore, only the configuration according to the present invention is described.

**[0032]** Content server 10 stores Java (registered trademark) AP software. Java AP software is software for enabling an application to be executed, and comprises a Java Archive (JAR) file containing a program written in the Java programming language and an Application Descriptor File (ADF) where information on the JAR file is described.

**[0033]** Content server 10 also stores an explanatory file associated with Java AP software stored therein. The contents of an explanatory file are shown in Fig. 3. As shown in Fig. 3, in a case where Java AP software is Java AP software for trial use (hereinafter, referred to as "a trial AP"), it is property is specified in an

anchor tag (tag beginning with “<A”). When button BT2 shown in Fig. 4 is clicked, an object tag (tag beginning with “<OBJECT”) is specified whose id property includes the value specified as the ilet property in the anchor tag. A URL (<http://WWW.CCC.co.jp/cardgame/jam>) specified as the data property of the object tag indicates a location for storing an ADF constituting the trial AP. It should be noted that, in a case where the Java AP software is not a trial AP, ijam property is specified in the anchor tag instead of ilet property. The explanatory file is written so as to provide the explanatory page shown in Fig. 4 when interpreted and executed by mobile phone 40.

**[0034]** Content server 10 also has a function of sending Java AP software to mobile phone 40. Content server 10, upon receiving from mobile phone 40 an HTTP request message requesting Java AP software, reads an ADF constituting the Java AP software. Content server 10 then generates an HTTP response message including the read ADF, and sends it to mobile phone 40. Content sever 10, upon receiving from mobile phone 40 an HTTP request message requesting the JAR file corresponding to the ADF, reads the JAR file. Content server 10 then generates an HTTP response message including the read JAR file, and sends it to mobile phone 40.

**[0035]** Following is a description of an ADF.

**[0036]** An ADF is a text file where a variety of control information is written for controlling the functions of storage, start, and network access, etc. of a JAR file. The data configuration of an ADF is shown in Fig. 5. In the “Item Name” column of Fig. 5, names of control information are listed. In the “Essential/Option” column, information indicating whether essential or optional for an ADF is listed. In the “Data” column, information on control information is listed.

**[0037]** Following is a brief description of each item of control information according to the embodiment. “AppName” is control information indicating the name of Java AP software, and information indicating the name is shown in the “Data” column of “AppName”. “PackageURL” is control information indicating the Uniform Resource Locator (URL) of a location accessed for downloading a

JAR file, and the URL of a location accessed for downloading a JAR file is shown in the “Data” column of “PackageURL”. “AppSize” is control information indicating the size of data of a JAR file, and the size of data is shown in the “Data” column of “AppSize”.

**[0038]** “IletPreserve” is storage control information indicating that Java AP software cannot be stored in mobile phone 40. In a case where an ADF contains “IletPreserve”, the ADF cannot be stored. On the other hand, in a case where an ADF does not contain “IletPreserve”, the ADF can be stored.

**[0039]** The above control information is set by content provider providing Java AP software. Description is made later of an environment where the Java AP software stated above is executed in mobile phone 40.

**[0040]** Configuration of Mobile Phone 40:

Following is a description of the configuration of mobile phone 40 with reference to Fig. 6. The configuration of mobile phone 40 is similar to that of common computers, therefore, only the configuration according to the present invention is described.

**[0041]** CPU 400 controls components of mobile phone 40 by executing a variety of programs stored in storage unit 406.

**[0042]** Storage unit 406 comprises Read Only Memory (ROM) 407, Random Access Memory (RAM) 408, and nonvolatile memory 409 such as an Electrically Erasable Programmable ROM (EEPROM).

**[0043]** ROM 407 stores programs such as an operating system for mobile phone 40, World Wide Web (WWW) browser software executed for downloading Java AP software, and software for establishing a runtime environment of Java AP software described later.

**[0044]** Nonvolatile memory 409 stores Java AP software, a variety of programs, and a variety of data. In nonvolatile memory 409, cache area 409a and storage area 409b are reserved for storing Java AP software.

**[0045]** Cache area 409a stores Java AP software downloaded from content server 10 for trial use in mobile phone 40. Java AP software stored in cache area 409a, upon exit from a program executed by CPU 400 for downloading the



software, is deleted by CPU 400 from cache area 409a. It should be noted that deletion of Java AP software specifically means that a memory area for storing the software is cleared, therefore it becomes impossible to read the software, and the memory area becomes available for storing new data.

5       **[0046]**     Storage area 409b stores Java AP software downloaded from content server 10, storage in mobile phone 40 which is instructed on the basis of storage control information and an instruction from a user. Java AP software stored in storage area 409b, unless a user of mobile phone 40 directs the deletion of the software, remains stored in storage area 409b.

10       **[0047]**     Nonvolatile memory 409 also stores a program to be executed when CPU 400 performs a speedy execution process. The speedy execution process is carried out as described below. CPU 400, when acquiring an ADF constituting Java AP software from content server 10, determines whether ilet property is specified in an anchor tag indicating the URL of the location for storing the ADF.  
 15       When it is determined that ilet property is specified in the anchor tag, CPU 400 identifies an object tag whose id property includes the value specified as the ilet property in the anchor tag, and extracts the URL specified as data property of the object tag. CPU 400 then requests content server 10 to send the ADF specified by the URL. CPU 400, upon receiving the ADF, stores it in cache area 409a, and  
 20       requests a server, accessed for downloading a JAR file specified by data of “PackageURL” of the ADF, to send the JAR file. CPU 400, upon receiving the JAR file from the server, stores it in cache area 409a, and executes a trial AP comprising the ADF and the JAR file without waiting for an instruction from a user.

25       **[0048]**     Nonvolatile memory 409 also stores a program to be executed when CPU 400 performs a storage process. The storage process is described below. CPU 400, after performing the aforementioned speedy execution process of a trial AP, determines whether an ADF constituting the trial AP contains “IletPreserve”. When it is determined that the ADF does not contain “IletPreserve”, CPU 400  
 30       determines the size of data that can be stored in a free space of storage area 409b, and compares the size of the free space with the total size of the data of the JAR

file specified by data of “AppSize” of the ADF and the size of the data of the ADF (hereinafter, referred to as “the size of an AP to be stored”). When it is determined that the size of the free space is equal to or more than the size of the AP to be stored, CPU 400 stores the trial AP in storage area 409b. When it is determined that the size of the free space is smaller than the size of the AP to be stored, CPU 400 lets a user select Java AP software to be deleted out of Java AP software stored in storage area 409b. When it is determined that the total size of the free space and the size of the data of selected Java AP software is equal to or more than the size of the AP to be stored, CPU 400 stores the trial AP in storage area 409b.

**[0049] Java Runtime Environment:**

Fig. 7 is a diagram showing a runtime environment of Java AP software stored in mobile phone 40. As software for establishing the runtime environment of Java AP software, mobile phone comprises the K Virtual Machine (KVM) and the Java 2 Micro Edition (J2ME) composed of the Connected Limited Device Configuration (CLDC) as a configuration and an original Java extended profile developed by a communication carrier as a profile.

**[0050]** A Java Application Manager (JAM) is software for causing CPU 400 to manage Java AP software stored in mobile phone 40 under the control of an operating system. CPU 400, in accordance with instructions from the JAM, carries out the functions of displaying a list of Java AP software stored in storage area 409b, of managing execution (start or forced termination, etc.) of Java AP software, of storing or modifying Java AP software, and of deleting Java AP software stored in mobile phone 40.

**[0051]** As a result of executing the above software by CPU 400, a runtime environment for Java AP software is established. In such a runtime environment, when CPU 400 executes Java AP software, the function of the software is realized.

**[0052] 2. Operation**

Following is a description of operations of the embodiment with reference to the drawings.

[0053] Fig. 8 is a flowchart illustrating a speedy execution process performed by CPU 400 of mobile phone 40.

[0054] When a user instructs WWW browser software to start by operating operation input unit 402 of mobile phone 40, CPU 400 executes the WWW browser software. CPU 400 performs operations in accordance with instructions from the WWW browser software, as described below. When a user accesses content server 10 by inputting the URL of content server 10, CPU 400 starts to communicate with content server 10. When a list file (not shown) is received from content server 10, CPU 400 interprets and executes the list file, and causes liquid crystal display unit 405 to display the list page as shown in Fig. 2. The list page of Fig. 2 lists names of Java AP software. When the user clicks button BT1 on the list page indicating the name of desired Java AP software using operation input unit 402 (in this operation example, it is assumed that the item "Card Game" is clicked), CPU 400 detects the click of button BT1, requests content server 10 to send the explanatory file as shown in Fig. 3 corresponding to the Java AP software. When the explanatory file is received from content server 10, CPU 400 interprets and executes the explanatory file, and causes liquid crystal display unit 405 to display the explanatory page as shown in Fig. 4.

[0055] When the user clicks button BT2 on the explanatory page using operation input unit 402, CPU 400 detects the click of button BT2 (Step S10), and determines whether ilet property is specified in the anchor tag described in the explanatory file corresponding to button BT2 (Step S11). When it is assumed that ilet property is specified in the anchor tag, CPU 400 determines that the Java AP software is a trial AP. CPU 400 identifies an object tag whose id property includes the value data specified as the ilet property in the anchor tag, and extracts the URL specified as data property of the object tag. CPU then sends an HTTP request message requesting the ADF specified by the URL to content server 10 via mobile packet communication network 30 and Internet 20 (Step S12).

[0056] Content server 10, upon receiving the HTTP request message, reads the ADF requested by the message. Content server 10 generates an HTTP response message including the read ADF, and sends it to mobile phone 40.

**[0057]** CPU 400 of mobile phone 40, upon receiving the HTTP response message (Step S13), on the basis of the contents of the ADF included in the message, determines whether it is possible to download Java AP software comprising the ADF. In this operation example, it is assumed that mobile phone  
 5 40 is capable of downloading the Java AP software; therefore, CPU 400 determines that it is possible to download the Java AP software. CPU 400 stores the ADF in cache area 409a temporarily (Step S14). CPU 400 extracts the PackageURL of the ADF; generates an HTTP request message requesting the JAR file specified by the PackageURL; and sends it to content server 10 (Step S15).

10 **[0058]** Content server 10, upon receiving the HTTP request message, generates an HTTP response message including the JAR file requested by the request message, and sends it to mobile phone 40.

**[0059]** CPU 400 of mobile phone 40, upon receiving the HTTP response message (Step S16), stores the JAR file included in the message in cache area  
 15 409a temporarily (Step S17).

**[0060]** CPU 400 executes the Java AP software composed of the ADF and the JAR file immediately without waiting for an instruction from the user (Step S18). Then, the function of the Java AP software is realized in mobile phone 40.

20 **[0061]** When it is determined in Step 11 that ilet property is not specified in the anchor tag, CPU 400 stores the Java AP software in storage area 409b (Step S19). CPU 400 carries out the processes of Steps S20 and S21, which are similar to those of Steps S15 and S16, and stores the received JAR file in cache area 409b (Step S22).

25 **[0062]** After the process of Step 18, when the user directs termination of the running Java AP software using operation input unit 402 of mobile phone 40, CPU 400 terminates the running software.

30 **[0063]** CPU 400 determines by referring to the ADF of the Java AP software whether the ADF does not contain "IletPreserve" (Step S30). When it is determined that the ADF does not contain "IletPreserve"; namely, that the Java AP software can be stored in mobile phone 40, CPU 400 causes liquid crystal display unit 405 to display a message saying "Is an application stored?" (Step S31).

**[0064]** When the user directs storage of the Java AP software by operating operation input unit 402 of mobile phone 40, CPU 400 detects the operation (Step S32), and reads from the ADF constituting the software the size of the data of the JAR file indicated by data of “AppSize” of the ADF. CPU 400 compares the size of the data of the JAR file with the size of the free space of storage area 409b (Step S33). When it is determined that the size of the data of the JAR file is smaller than the size of the free space, CPU 400 stores the Java AP software in cache area 409b (Step S34).

**[0065]** When it is determined in Step S33 that the size of the data of the JAR file is more than the size of the free space, CPU 400 carries out the space freeing process, as described below (Step S35).

**[0066]** Fig. 10 is a flowchart illustrating the space freeing process.

**[0067]** CPU 400, in accordance with instructions from the JAM, causes liquid crystal display unit 405 to display a list of Java AP software stored in storage area 409b, and a message saying “If you delete any of Java AP software on the list, you can store this Java AP software” (Step S350).

**[0068]** When the user selects the Java AP software to be deleted, by operating operation input unit 402 of mobile phone 40, CPU 400 detects the operation (Step S351), and determines the size of the data of the selected software. CPU 400 calculates the total size of the data and the size of the free space of storage area 409b (hereinafter, referred to as “the total size”) (Step S352). CPU 400 then compares the total size with the size of the data of the JAR file (Step S353). When it is determined that the total size is equal to or more than the size of the data of the JAR file, CPU 400 causes liquid crystal display unit 405 to display the name of the selected Java AP software to be deleted, and a message asking “Is it ok to delete the Java AP software?” (Step S355).

**[0069]** When the user directs deletion of the Java AP software by operating operation input unit 402 of mobile phone 40, CPU 400 detects the operation (Step S356), and deletes the software from storage area 409b (Step S357). CPU 400 stores in cache area 409b the Java AP software to be stored (Step S358).

**[0070]** When it is determined in Step S353 that the total size is smaller than the size of the data of the JAR file, CPU 400 causes liquid crystal display unit 405 to display a message saying “The size of the free space is still insufficient. Please select other Java AP software to be deleted”. The processes of Steps S351 to S354 are repeated, until the total size of Java AP software to be deleted is equal to or more than the size of the data of the JAR file, or until CPU 400 detects that the user has directed the cancellation of the storage process in Step S354.

**[0071]** As described above, CPU carries out the storage process.

**[0072]** When it is determined in Step S30 in Fig. 9 that the ADF contains “IletPreserve”; namely, that the Java AP software cannot be stored in mobile phone 40, the storage process as described above is not carried out for the Java AP software.

**[0073]** When the user directs termination of the WWW browser software using operation input unit 402 of mobile phone 40, CPU 400 terminates the software, and deletes contents stored in cache area 409a.

**[0074]** When the user desires to execute Java AP software stored in storage area 409b, he/she, by operating operation input unit 402 of mobile phone 40, provides an instruction requesting liquid crystal display unit 405 to display a list of Java AP software stored in storage area 409b. CPU 400, upon receiving the instruction, causes liquid crystal display unit 405 to display a list of Java AP software stored in storage area 409b, in accordance with instructions from the JAM. When the user directs execution of the desired Java AP software, CPU 400 reads the software from storage area 409b, and executes it. Subsequently, the function of the Java AP software is realized in mobile phone 40.

**[0075]** Java AP software stored in storage area 409b remains stored in storage area 409b, unless the user deletes the software.

**[0076]** With the above configuration, a user can readily try Java AP software in mobile phone 40. After trying Java AP software, the user can also store the software in mobile phone 40 on the basis of the presence or the absence of storage control information and at the will of the user. In a case where the size of the free space of mobile phone 40 is insufficient for storing the Java AP software, the user

can store the software by deleting other Java AP software stored in mobile phone 40.

[0077] In the embodiment, information indicating that Java AP software is a trial AP is specified in an anchor tag of the explanatory file, which is referred to for downloading the ADF constituting the Java AP software. With the above configuration, it becomes unnecessary to modify the Java AP software itself when the software is set as a trial AP.

### [0078] 3. Modifications

In the foregoing, the embodiment of the present invention has been described. The embodiment is intended to be illustrative, and the present invention can be implemented in various other embodiments without departing from the main characteristic of the invention. For example, modifications such as the following are possible.

#### [0079] Modification 1:

In the above embodiment, content server 10 may be directly connected via a dedicated line instead of Internet 20, with gateway server 31 in mobile packet communication network 30. Content server 10 also may be provided within mobile packet communication network 30. Gateway server 31 may have the function of content server 10.

[0080] In the above embodiment, files may be exchanged using the HTTPS, instead of the HTTP, to assure higher security.

#### [0081] Modification 2:

In the above embodiment, Java AP software written in the Java programming language is treated as contents. However, a programming language is not limited to the Java programming language, and may be the C++ language, etc.

[0082] Contents may be image data, video data, music data, or document data such as HTML data, instead of software. To enable mobile phone 40 to use the above contents, mobile phone 40 is configured to comprise a program and a unit for processing image data, video data, or music data, and a program for interpreting and displaying document data in liquid crystal display unit 405.

Mobile phone 40 is also configured to receive from content server 10, information on the size of the data of contents, which is referred to in the storage process of the above embodiment, before downloading the contents.

**[0083]**     Modification 3:

5             In the above embodiment, Java AP software is downloaded from content server 10 to mobile phone 40; however, Java AP software may be delivered from content server 10 to mobile phone 40.

**[0084]**     Modification 4:

10            In the above embodiment, in nonvolatile memory 409 of mobile phone 40, cache area 409a and storage area 409b may not be assigned. CPU 400, when storing in nonvolatile memory 409 Java AP software downloaded from content server 10, associates with the software storage identification data (for example, a memory flag "0") indicating that the software is stored temporarily. In the storage process as shown in Fig. 9, when a user directs storage of Java AP software, the  
15            software is not deleted from nonvolatile memory 409, but the storage identification data corresponding to the software is modified to storage identification data indicating that the software is stored enduringly (for example, the memory flag is modified from "0" to "1"). On the other hand, when the user does not direct storage of the software, after a running WWW browser software is  
20            terminated, the software associated with the memory flag "0" is deleted from nonvolatile memory 409.

**[0085]**     With the above configuration, areas of nonvolatile memory 409 for storing Java AP software can be used effectively.

**[0086]**     Modification 5:

25            In the above embodiment, Java AP software is associated with trial information indicating that the software is contents for trial use, by specifying ilet property in an anchor tag described in an explanatory file. However, it is possible that Java AP software is not associated with such trial information. In this case, CPU 400 of mobile phone 40, after downloading Java AP software from content  
30            server 10, stores the software in cache area 409a, and performs the aforementioned speedy execution process of the software.



**[0087] Modification 6:**

Software such as JAM software or operating system software executed by CPU 400 of mobile phone 40 in the above embodiment can be provided to mobile phone 40 through a CPU 400 readable storage medium storing the software such as a magnetic recording medium, a magneto-optical recording medium, or ROM. The software also can be downloaded to mobile phone 40 via Internet 20.